

CLAIMS

What is claimed is:

1 1. A method comprising:
2 a first module of a thread in a computing environment dynamically causing
3 a second module to be logically chained to itself, enabling the second module to
4 be executed as part of the same thread;
5 causing the second module dynamically and logically chained as part of
6 the same thread to register first events of interest to the second module with an
7 event notification service of the computing environment;
8 waiting for notification of occurrence of one or more of the first events; and
9 causing the second module dynamically and logically chained as part of
10 the same thread to process an occurred one of the first events.

1 2. The method of claim 1, wherein
2 the second module comprises a first set of executable instructions
3 designed to register the first events with said event notification service, and a first
4 pointer to the first set of executable instructions;
5 the first module dynamically causing the second module to be logically
6 chained to itself, by invoking a thread module chaining service of the computing
7 environment; and
8 the method further comprises the thread module chaining service
9 annotating a control data structure of the thread to logically associate the second
10 module with the first module, including with the annotation, the first pointer to the
11 first set of executable instructions of the second module.

1 3. The method of claim 2, wherein

2 said causing of the second module dynamically and logically chained as
3 part of the same thread to register first events of interest to the second module
4 with an event notification service of the computing environment comprises the
5 first module invoking the thread module chaining service to orchestrate
6 registration of events of interest to logically chained modules of the thread with
7 the event notification service; and

8 the method further comprises the thread module chaining service
9 retrieving the first pointer to the first set of executable instructions, and causing
10 the first set of executable instructions to be executed, using the first pointer to
11 locate the first set of executable instructions.

1 4. The method of claim 3, wherein

2 the second module further comprises a second set of executable
3 instructions designed to process an occurred one of the first events, and a
4 second pointer to the second set of executable instructions; and

5 the thread module chaining service further includes with the annotation,
6 the second pointer to the second set of executable instructions of the second
7 module.

1 5. The method of claim 4, wherein

2 said causing of the second module dynamically and logically chained as
3 part of the same thread to process an occurred one of the first events comprises
4 the first module invoking the thread module chaining service to orchestrate
5 processing of an occurred event by the logically chained modules of the thread;
6 and

7 the method further comprises the thread module chaining service
8 retrieving the second pointer to the second set of executable instructions, and

9 causing the second set of executable instructions to be executed, using the
10 second pointer to locate the second set of executable instructions.

1 6. The method of claim 5, wherein the method further comprises
2 detecting for a thread termination condition; and
3 causing the thread to be terminated after detecting the thread termination
4 condition.

1 7. The method of claim 6, wherein
2 the second module further comprises a third set of executable instructions
3 designed to perform termination clean up for the second module, and a third
4 pointer to the third set of executable instructions;
5 the thread module chaining service further includes with the annotation,
6 the third pointer to the third set of executable instructions of the second module;
7 said detecting comprises the first module detecting for the thread
8 termination condition;
9 said causing of the thread to be terminated after detecting the thread
10 termination condition comprises the first module invoking the thread module
11 chaining service to orchestrate thread termination clean up by the logically
12 chained modules of the thread; and
13 the method further comprises the thread module chaining service
14 retrieving the third pointer to the third set of executable instructions, and causing
15 the third set of executable instructions to be executed, using the third pointer to
16 locate the third set of executable instructions.

1 8. The method of claim 1, wherein the method further comprises

2 the first module dynamically causing a third module to be logically chained
3 to the first and second modules, enabling the third module to be executed as part
4 of the same thread;

5 causing the third module dynamically and logically chained as part of the
6 same thread to register second events of interest to the third module with the
7 event notification service of the computing environment; and

8 causing the dynamically and logically chained third module as part of the
9 same thread to process an occurred one of the second events, said waiting for
10 notification of occurrence further comprising waiting for notification of one or
11 more of the second events.

1 9. A method comprising:

2 a first module of a thread in a computing environment dynamically causing
3 a second module to be logically chained to itself, enabling the second module to
4 be executed as part of the same thread;

5 waiting for notification of occurrence of one or more of first events of
6 interest to the second module dynamically and logically chained as part of the
7 same thread;

8 causing the second module dynamically and logically chained as part of
9 the same thread to process an occurred one of the first events;

10 detecting for a thread termination condition; and

11 causing the thread to be terminated after detecting the thread termination
12 condition.

1 10. The method of claim 9, wherein

2 the second module comprises a first set of executable instructions
3 designed to process an occurred one of the first events, and a first pointer to the
4 first set of executable instructions;

5 the method further comprises a thread module chaining service annotating
6 a control data structure of the thread to logically associate the second module
7 with the first module, including with the annotation, the first pointer to the first set
8 of executable instructions of the second module;

9 said causing of the second module dynamically and logically chained as
10 part of the same thread to process an occurred one of the first events comprises
11 the first module invoking the thread module chaining service to orchestrate
12 processing of an occurred event by logically chained modules of the thread; and

13 the method further comprises the thread module chaining service
14 retrieving the first pointer to the first set of executable instructions, and causing
15 the first set of executable instructions to be executed, using the first pointer to
16 locate the first set of executable instructions.

1 11. The method of claim 10, wherein

2 the second module further comprises a second set of executable
3 instructions designed to perform termination clean up for the second module, and
4 a second pointer to the second set of executable instructions;

5 the thread module chaining service further includes with the annotation,
6 the second pointer to the second set of executable instructions of the second
7 module;

8 said detecting comprises the first module detecting for the thread
9 termination condition;

10 said causing of the thread to be terminated after detecting the thread
11 termination condition comprises the first module invoking the thread module

12 chaining service to orchestrate thread termination clean up by the logically
13 chained modules of the thread; and
14 the method further comprises the thread module chaining service
15 retrieving the second pointer to the second set of executable instructions, and
16 causing the second set of executable instructions to be executed, using the
17 second pointer to locate the second set of executable instructions.

1 12. The method of claim 9, wherein
2 the second module comprises a first set of executable instructions
3 designed to perform termination clean up for the second module, and a first
4 pointer to the first set of executable instructions;
5 the method further comprises a thread module chaining service annotating
6 a control data structure of the thread to logically associate the second module
7 with the first module, including with the annotation, the first pointer to the first set
8 of executable instructions of the second module;
9 said detecting comprises the first module detecting for the thread
10 termination condition;
11 said causing of the thread to be terminated after detecting the thread
12 termination condition comprises the first module invoking the thread module
13 chaining service to orchestrate thread termination clean up by logically chained
14 modules of the thread; and
15 the method further comprises the thread module chaining service
16 retrieving the first pointer to the first set of executable instructions, and causing
17 the first set of executable instructions to be executed, using the first pointer to
18 locate the second set of executable instructions.

1 13. The method of claim 9, wherein the method further comprises

2 the first module dynamically causing a third module to be logically chained
3 to the first and second modules, enabling the third module to be executed as part
4 of the same thread; and

5 causing the third module dynamically and logically chained as part of the
6 same thread to process an occurred one of second events, said waiting for
7 notification of occurrence of one or more of first events further comprising waiting
8 for notification of one or more of the second events.

1 14. A computing device comprising:

2 storage medium having stored therein a first plurality of executable
3 instructions designed to provide a thread module chaining service to facilitate
4 dynamic logical chaining of a plurality of modules to execute together as parts of
5 a single thread on the computing device, including an ability to maintain and
6 annotate a thread control data structure with control data to enable said logical
7 chaining of the modules and their orchestrated execution as parts of a single
8 thread; and

9 at least one processor coupled to the storage medium to execute the
10 instructions.

1 15. The computing device of claim 14, wherein the ability to maintain and
2 annotate a thread control data structure with control data to enable said logical
3 chaining of the modules and their orchestrated execution as parts of a single
4 thread includes an ability to annotate the thread control data structure with
5 control data about a module to be logically chained to be a part of a thread, when
6 dynamically invoked to logically chain the module to be a part of the thread.

1 16. The computing device of claim 15, wherein

2 the control data includes at least a selected one of
3 a pointer of the module pointing to a plurality of executable instructions
4 of the module to register events of interest to the module with an
5 event notification service of the computing device,
6 a pointer of the module pointing to a plurality of executable instructions
7 of the module to process an occurred event of interest to the
8 module,
9 a pointer of the module pointing to a plurality of executable instructions
10 of the module to perform thread termination clean up for the
11 module, and
12 the ability includes an ability to extract the selected one or more pointers
13 from the module.

1 17. The computing device of claim 14, wherein the first plurality of executable
2 instructions further provide the thread module chaining service with at least a
3 selected one of
4 an ability to orchestrate registration of events of interest to the logically
5 chained modules with an event notification service by the logically
6 chained modules,
7 an ability to orchestrate processing of an occurred event of interest by one
8 or more of the logically chained modules, and
9 an ability to orchestrate thread termination clean up of the logically
10 chained modules by the respective logically chained modules.

1 18. The computing device of claim 14, wherein the computing device
2 comprises a UPNP control point.

1 19. A computing device comprising:
2 storage medium having stored therein a first and a second module, with
3 the first module equipped to logically chain the second module to the first module
4 dynamically during execution of the first module, enabling the second module to
5 execute with the first module as a single thread, and the second module having
6 at least a selected one of a set of executable instructions to register events of
7 interest to the second module, a set of executable instructions to process an
8 occurred one of the events of interest, and a set of executable instructions to
9 perform clean up during thread termination; and
10 at least one processor coupled to the storage medium to execute the
11 instructions.

1 20. The computing device of claim 19, wherein the second module further
2 includes at least a corresponding one of a first point, a second pointer, and a
3 third pointer pointing to the first set, the second set, and the third set of
4 executable instructions respectively.

1 21. The computing device of claim 19, wherein the computing device
2 comprises a UPNP control point.

1 22. An article of manufacture comprising:
2 a computer readable medium; and
3 a plurality of executable instructions designed to implement a thread
4 module chaining service to facilitate dynamic logical chaining of a plurality of
5 modules to execute together as parts of a single thread in a computing
6 environment, including an ability to maintain and annotate a thread control data

7 structure with control data to enable said logical chaining of the modules and
8 their orchestrated execution as parts of a single thread.
9

1 23. The article of claim 22, wherein the ability to maintain and annotate a
2 thread control data structure with control data to enable said logical chaining of
3 the modules and their orchestrated execution as parts of a single thread includes
4 an ability to annotate the thread control data structure with control data about a
5 module to be logically chained to be a part of a thread, when dynamically invoked
6 to logically chain the module to be a part of the thread.
7

1 24. The article of claim 22, wherein the first plurality of executable instructions
2 further provide the thread module chaining service with at least a selected one of
3 an ability to orchestrate registration of events of interest to the logically
4 chained modules with an event notification service by the logically
5 chained modules,
6 an ability to orchestrate processing of an occurred event of interest by one
7 or more of the logically chained modules, and
8 an ability to orchestrate thread termination clean up of the logically
9 chained modules by the respective logically chained modules.
10

1 25. A system comprising:
2 a first device including a plurality of executable instructions designed to
3 implement a thread module chaining service to facilitate dynamic logical chaining
4 of a plurality of modules to execute together as parts of a single thread in the first

5 device, including an ability to maintain and annotate a thread control data
6 structure with control data to enable said logical chaining of the modules and
7 their orchestrated execution as parts of a single thread; and
8 a second device coupled to the first device.

1 26. The system of claim 25, wherein the ability of the thread module chaining
2 service to maintain and annotate a thread control data structure with control data
3 to enable said logical chaining of the modules and their orchestrated execution
4 as parts of a single thread includes an ability to annotate the thread control data
5 structure with control data about a module to be logically chained to be a part of
6 a thread, when dynamically invoked to logically chain the module to be a part of
7 the thread.

1 27. The system of claim 25, wherein the plurality of executable instructions
2 further provide the thread module chaining service with at least a selected one of
3 an ability to orchestrate registration of events of interest to the logically
4 chained modules with an event notification service by the logically
5 chained modules,
6 an ability to orchestrate processing of an occurred event of interest by one
7 or more of the logically chained modules, and
8 an ability to orchestrate thread termination clean up of the logically
9 chained modules by the respective logically chained modules.

1 28. The system of claim 25, wherein the first computing device comprises a
2 UPNP control point, and the second computing device comprises a selected one
3 of a UPNP server and a UPNP renderer.

1 29. A system comprising:

2 a first device having a first and a second module, with the first module
3 equipped to logically chain the second module to the first module dynamically
4 during execution of the first module, enabling the second module to execute with
5 the first module as a single thread, and the second module having at least a
6 selected one of a set of executable instructions to register events of interest to
7 the second module, a set of executable instructions to process an occurred one
8 of the events of interest, and a set of executable instructions to perform clean up
9 during thread termination; and
10 a second device coupled to the first device.

1 30. The system of claim 29, wherein the second module further includes at
2 least a corresponding one of a first point, a second pointer, and a third pointer
3 pointing to the first set, the second set, and the third set of executable
4 instructions respectively, .

1 31. The system of claim 29, wherein the first computing device comprises a
2 UPNP control point, and the second computing device comprises a selected one
3 of a UPNP server and a UPNP rendered.